IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MUSIC RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING AND K-MEANS CLUSTERING

[1]Arunava Mukhopadhyay, [2]Udita Chakraborty, [2]Debkanya Banerjee, [1]Arindam Chakraborty

[1]Assistant Professor, [2]Student

Electronics and Communication Engineering Department

Institute of Engineering and Management, University of Engineering and Management, Kolkata, India

Abstract: The Music Recommendation System is a machine learning-based approach that enables the music provider to anticipate customer preferences and recommend relevant songs based on the properties of previously heard music. This proposed system revolves around two major components, i.e., user data and content data. The user data consists of various factors like listening history, ambiance, time, genre, etc. After considering all the factors, a pathway is created to recommend the songs accordingly. To build the proposed system, we have to import certain libraries such as NumPy, Pandas, Matplotlib, and Seaborn, which are data processing and visualization libraries. It is required to have selection of similarity metric like cosine similarity, then the model must score each candidate according to this similarity metric and thereafter the system will recommend according to this score. In this work, mainly implementation of a Collaborative Filtering Mechanism is done thus, the system will only be able to make recommendations based on that specific user's interests. Using the K-means clustering algorithm, suggestion of music can be performed that is similar to a user's preferences even if they enjoy different genres. For our proposed model, chosen input is the respective dataset, and the output is the recommended songs.

***Index Terms* - Machine Learning, Cosine Similarity, Collaborative Filtering, K-means Clustering Algorithm.**

I. INTRODUCTION

A rapid-fire growth in online and mobile platforms are observed extensively in today's world and lots of music platforms are coming into the picture. These platforms are offering song lists from across the globe. Every existent has a unique taste for music. Online Music listeners have lots of choices for the song. These customers occasionally face difficulty while selecting the songs or browsing the long list. The service providers need an effective and accurate recommender system for suggesting applicable songs. In this proposed system, we present a music system, which provides an individualized service of music recommendation. In order to build a personalized recommendation system., a few questions need to be answered first, namely: how to compute the property of the songs presented in the dataset, how to analyze the preference of the users and how to select which music gets recommended to which user. In this work each of these questions are answered. Here, we compute the property of songs using a similarity metric like cosine similarity and make a cluster of analogous songs using K-means clustering algorithm. The user's preference is detected when the user provides us with the song title based on which they want their music recommendations to be. Lastly, assigning songs to users based on collaborative filtering technique is executed. Based on the levels of the users' musical preferences and loves, a collaborative filtering-based suggestion technique is suggested. In order to create a prediction, collaborative filtering focuses on the connections between people and items. The thing of the recommender system is to reckon a scoring function that summations the result of calculating parallels between users and between particulars. First, the user- item standing matrix is used to form stoner clusters and item clusters. Next, these clusters are used to find the most analogous user cluster or most analogous item cluster to a target user. Eventually, songs are recommended from the most analogous user and item clusters. It determines which features are most important for suggesting the songs. We're using a Spotify dataset to find similar songs for recommendation using cosine similarity. Certain libraries like NumPy, Pandas, Matplotlib, and Seaborn, which are data processing and visualization libraries are also incorporated. K- Means Clustering Algorithm is chosen to apply for this work. By using this model, people may find new interests. Separately, the machine learning system could not be aware that the user is interested in a certain item, but the model may nevertheless suggest it since similar people also seem to be interested in it.

D Kim et al. (2020) presented a method for personalized music services using a dynamic K-means clustering algorithm with a focus on analysing users' preferences using Shortest Time Fourier Transform (STFT) to extract music properties from sound waves and save the list of music the user downloaded or listened to infer their preferences [1]. P. Cano et al. (2005) discussed content-based system in literature. They extracted descriptions of instrumentation, rhythm, and harmony from music signals using similarity criteria [2]. In order to enhance collaborative filtering recommender systems that use ratings, Mehdi Srifi et al. offer a thorough survey of recent studies that incorporates review texts. The three main approaches used by the authors of this research to classify existing reviews are those that are based on words, subjects, and attitudes. The writers go into how user review texts have been used for each system to enhance rating profiles and determine feature preference. [3]. B. Logan offered that advice based only on song properties as extracted using MFCC (Mel Frequency Cepstral Coefficients), which creates 13-dimensional vectors from groupings of related songs called "song sets" [4]. Various recommendation systems that are now in use, including content-based, collaborative, emotion-based, and other strategies, are reviewed by Paul and Kundu (2020), along with a discussion of their advantages and disadvantages. A hybrid recommendation system is also built in this paper in order to solve the challenges in the existing recommender systems. This hybrid system uses the principle of both content-based and collaborative filtering systems while recommending song to the users [5]. In a study by Specht et al. (2017) recommendation approach is proposed which aims to provide track recommendations for a given user in each context using Factorization Machines (FM). Then, a comparison is drawn between this FM based approach and three baseline recommender systems: a CF-based and SVD-based system, and a classification-based system through a series of experiments [6].

There are many recommender systems prevalent in the market but none of which combines Collaborative filtering along with K-means clustering algorithm and also incorporates similarity metric like cosine similarity unlike our model.

II. SET UP AND IMPLEMENTATION

A. Collaborative Filtering

Collaborative filtering makes recommendations based on how similar users or products are to one another. The algorithm's fundamental presumption is that people with comparable interests will share similar preferences. We are aware of two different kinds of recommender systems; the content-based systems have fewer use cases and a higher temporal complexity. Additionally, unlike algorithms based on collaborative filtering, this algorithm is based on a restricted amount of content. These recommender systems' ability to deliver personalized material effectively while also being able to adjust to shifting preferences is one of their key features. Collaborative filtering has two prominent classes: item-based and user-based. Item-based evaluates similarity between items based on the primary users' ratings or interactions with other items. User-based measures similarity between the main user and other users. Figure 1 (a) and Figure 1 (b) below represent the respective images of the classes.

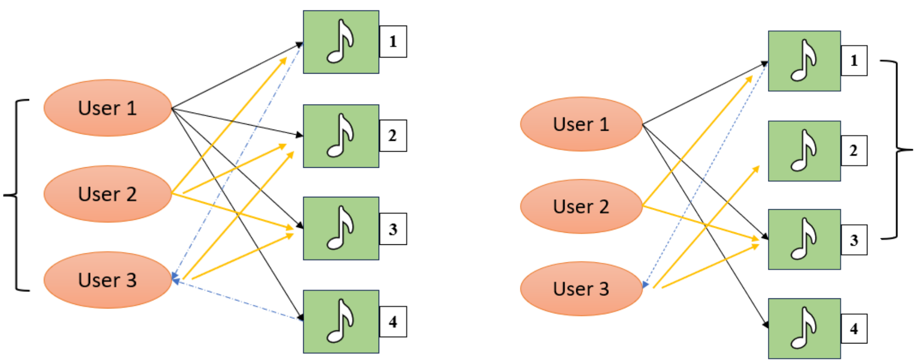

Figure 1. (a) User based filtering, (b) Item based filtering

B. Cosine Similarity

It is equal to the cosine of the angle between two vectors. This can be used to measure the distance between two locations in a specific plane. This simply relies on the cosine principle, according to which the similarity of data points decreases with increasing distance. Similar to cosine angles, cosine similarity is used in the recommendation system. The least suggested content will be that with the lowest degree of similarity, and the most recommended content will be that with the highest degree of similarity. Figure 2 reflects the basic mechanism based for cosine similarity.

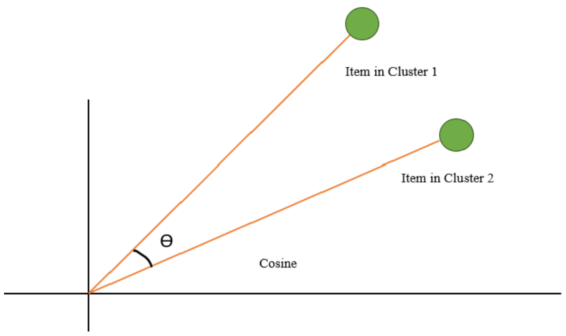

Figure 2. Cluster-based cosine similarity

For textual data, it is also important to compare the vectorized texts from the original text source. There are many ways to compare two pieces of content, and recommendation systems frequently employ the similarity matrix to make relevant content suggestions to users depending on their accessing abilities. As a result, it is possible to collect any recommendation data, and from that data, the features that are important for recommending the contents can be retrieved. The textual data must be vectorized using the Count-Vectorizer after it is made accessible in order to obtain the similarity matrix. The cosine similarity metrics of sci-kit learn can be used to make appropriate recommendations to the user after the similarity matrix has been created. For the specified textual data for recommendations, the cosine similarity would therefore offer a similarity matrix, and the content with greater similarity scores can be ordered using lists. The cosine similarity in this system would take into account the frequently occurring phrases in the textual data, vectorize those terms with higher frequencies, and recommend that material with higher recommendation percentages. It serves as a distance measurement metric between two points in a plane and is equal to the cosine of the angle formed by two vectors. This solely relies on the cosine principle, which states that as distance increases, the similarity of data points decreases. Similar to cosine angles, cosine similarity is used in the recommendation system. If the similarity of the content is low, it will be considered the least recommended content, and if the similarity of the content is high, the generated recommendations will be at the top. Finding similarities between the vectorized texts from the original text document is also necessary for textual data. There are numerous ways to compare two pieces of content, and recommendation systems often use the similarity matrix to suggest related material to the user based on the user's accessing capabilities.

Therefore, any recommendation data can be obtained, and the necessary characteristics that are helpful for recommending the contents may be extracted from the data. The Count-Vectorizer must be used to vectorize the text data once it is available in order to create the similarity matrix. After obtaining the similarity matrix, sci-kit learn's cosine similarity metrics can be used to make appropriate user recommendations. For the specified textual data for recommendations, the cosine similarity would therefore offer a similarity matrix, and the content with greater similarity scores can be ordered using lists. The frequent terms in the textual data would be taken into account by this system's cosine similarity algorithm, which would vectorize those phrases with greater frequencies and propose that material with higher recommendation percentages.

C. CountVectorizer

A tool provided by the Python sci-kit-learn module turns a text that is supplied into a vector depending on how many times each word appears across the entire text. This is useful if we want to turn each word in each similar text into a vector and we have similar texts. Every word is represented by a column in the matrix, and each sample of text from the document is a row, forming a matrix. The value of each cell represents the number of words in that specific text sample.

D. Min-Max Scaler

A normalization approach called min-max scaling uses the minimum and maximum value of each feature to scale data in a dataset for a particular range. The min-max scaler employs the lowest and maximum values of each column to scale the data series as opposed to standard scaling, which scales data based on the normal distribution.

E. Pre-filtering Technique

Pre-filtering is a technique applied in recommendation systems to enhance the accuracy of recommendations by reducing the size of the dataset that is used to generate recommendations. In this, a subset of the dataset is selected based on some criteria, like popularity or relevance, and only this subset is used to induce recommendations. This can help to reduce the noise in the dataset and improve the accurateness of the recommendations. However, it might also limit the diversity of recommendations and might not be effective for all types of recommendations.

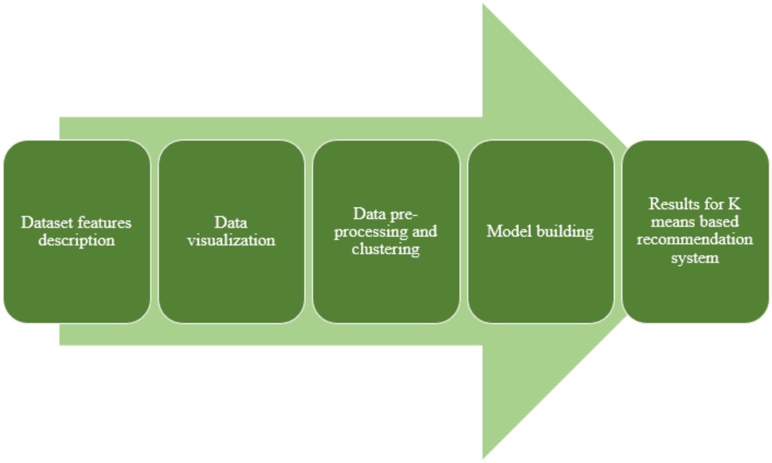

Figure 3. Process implementation with K means algorithm

F. K-Means Clustering

With K denoting the necessary number of pre-defined clusters, it is an unsupervised learning algorithm that divides the unlabeled dataset into clusters. The dataset is divided into k unique clusters using an iterative approach, and each cluster only includes datasets with similar characteristics. The major objective of this technique is to reduce the total distances between the data points and the clusters to which they are representing. In figure 3 depict steps for executing the recommendation model.

Figure 4 shows here the flow chart involved during the incorporated process to move with the recommendation system.

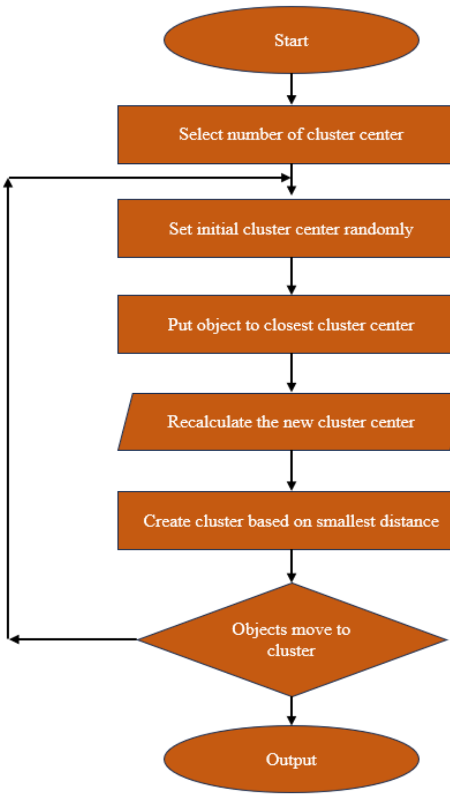

Figure 4. Flow chart of the involved process

III. RESULTS AND DISCUSSION

For our system, the input is the dataset, and the output is the recommended songs.

Test case 1:

Here, the system is instructed to produce a list of 14 such songs which has similar moods to the song "Mind Mischief" by 'Tame Impala'. Figure 5 and Figure 6 depict the Test Case 1 related input and output respectively.

```
recommendations = Spotify_Recommendation(data)
recommendations.recommend("Mind Mischief", 14)
```

Figure 5. Input for Test Case 1

	artists	name
156637	['Tame Impala']	Lucidity
55075	['Scissor Sisters']	Take Your Mama
48178	['Jefferson Airplane']	Somebody to Love - Live at The Woodstock Music...
56450	['Brad Paisley', 'Carrie Underwood']	Remind Me (with Carrie Underwood)
134591	['Bob Dylan']	Is Your Love in Vain?
104194	['Priscilla Chan']	夜機
136064	['Simply Red']	Heaven - 2008 Remaster
12763	['Iron Maiden']	Aces High - 2015 Remaster
11061	['Foghat']	Fool for the City
165416	['Elvis Presley']	He'll Have to Go
10415	['The Rolling Stones']	Rocks Off
122449	['Jessica Simpson', 'Nick Lachey']	Where You Are (featuring Nick Lachey) (feat. N...
70421	['New Order']	Round & Round - 2015 Remaster
12643	['Alabama']	The Closer You Get

Figure 6. Output for Test Case 1

Test Case 2:

In this instance the system is instructed to produce a list of 8 such songs which has similar moods to the son "Eclipse" by 'John Denver'. Figure 7 and Figure 8 show the Test Case 2 related input and output respectively.

```
recommendations = Spotify_Recommendation(data)
recommendations.recommend("Eclipse", 8)
```

Figure 7. Input for Test Case 2

	artists	name
8970	['Charmian Carr', 'Heather Menzies', 'Nicholas...	The Sound of Music
44077	['Cole Porter', 'Alfred Drake', 'Pembroke Dave...	Kiss Me, Kate: So in Love (Reprise)
6071	['Alexander Scriabin', 'Vladimir Horowitz']	Etude in D-Sharp Minor, Op. 8, No. 12: Patetic...
16700	['Madvillain', 'Madlib', 'MF DOOM']	All Caps
167446	['Percy Faith & His Orchestra']	The Holly and the Ivy / Here We Go A-Caroling ...
64114	['Frank Sinatra']	Let's Get Away From It All
99612	['Jefferson Airplane']	Blues from an Airplane
164029	['Smokey Robinson & The Miracles']	I Believe In Christmas Eve

Figure 8. Output for Test Case 2

Test Case 3:

In this case the system is instructed to produce a list of 10 such songs which has similar moods to the song "Please" by 'Anson Weeks and His Orchestra'. Figure 9 and Figure 10 show the Test Case 3 oriented input and output respectively.

```
recommendations = Spotify_Recommendation(data)
recommendations.recommend("Please", 10)
```

Figure 9. Input for Test Case 3

	artists	name
40855	['Dea Garbaccio', 'Aldo Donà', 'Nella Colombo']	Il tamburo della banda d'Affori
96293	['Geeta Dutt']	Rooth Gaye Hamse Piya
4102	['Benny Goodman', 'Peggy Lee']	How Deep Is the Ocean (feat. Peggy Lee)
146534	['Machito Orchestra']	Torero
96712	['Lionel Hampton and his Sextet']	I'll Remember April
55158	['Joy/Disaster']	Ecoute moi - Demo july 2004
129536	['Alpana Banerjee']	Aaji Ei Sandhyay
41092	['Richard Himber and his Orchestra', 'Joey Nash']	A Thousand Goodnights
110876	['Samuel Barber', 'Leontyne Price']	Hermit Songs, Op. 29: Saint Ita's Vision
42192	['Nedime Hanım']	Ümidini Kirpiklerine

Figure 10. Output for Test Case 3

Test Case 4:

In this instance the system is instructed to produce a list of 12 such songs which has similar moods to the song "I Put A Spell On You" by 'Screamin' Jay Hawkins'. Figure 11 and Figure 12 depict the Test Case 4 related input and output respectively.

```
recommendations = Spotify_Recommendation(data)
recommendations.recommend("I Put A Spell On You", 12)
```

Figure 11. Input for Test Case 4

	artists	name
170070	['Majority of One']	So Close
80254	['Umberto Giordano', 'Mario Rossi', 'Orchestra...	Amor ti vieta di non amar
63496	['George Jones']	Eskimo Pie
135837	['Antonio Aguilar']	El Hombre Alegre
62210	['Mohantara Talpade']	Hum Hind Ki Hai Naariyan
24826	['Franz Joseph Haydn', 'Isaac Stern', 'Mstisla...	London Trio No. 1 in C Major, Hob. IV: 1 - I. ...
168685	['Sting']	She's Too Good For Me
42704	['Claudio Villa']	Se mi vuoi bene baciami
22259	['Roy Fox']	Dinner at Eight
59889	['Johann Sebastian Bach', 'Albert Schweitzer']	Prelude and Fugue in C Major, BWV 545: I. Prelude
160523	['Christoph Willibald Gluck', 'Arturo Toscanini']	Orfeo ed Euridice, Wq. 30: Act II: Scene 2 - D...
5316	['Gene Autry']	Here Comes Santa Claus (Right Down Santa Claus...

Figure 12. Output for Test Case 4

The Music Recommendation System using Collaborative Filtering and K-means Clustering Algorithm with cosine similarity is a powerful approach. The system is based on collaborative filtering, which uses user ratings and preferences to predict which songs or artists a user may like. One of the significant advantages of using cosine similarity is that it measures the similarity between two vectors irrespective of their magnitude. This means that the algorithm can identify similarities between two users' listening preferences, even if they have rated different numbers of songs. The k-means clustering algorithm further groups similar songs together, makes it easier for the system to recommend related songs to the user.

However, there are some potential limitations to this approach. One of the primary challenges is the "cold start" problem, where the system struggles to provide recommendations for new users who have not yet rated any songs. Another limitation is that the system can struggle to recommend niche or less popular music genres that have limited user data and ratings. Additionally, the quality and size of the dataset used to train the system can have a significant impact on its effectiveness. Therefore, it is essential to consider the limitations and potential enhancements to this approach to further improve the system's accuracy and effectiveness.

IV. CONCLUSION

In this work, the presented model to recommend music based on the user's previous preferences. This work proposed, designed & developed a music recommendation system using collaborative filtering and k-means clustering algorithm. Here, a similarity score for each song is established based on the several attributes of the song such as liveness, danceability, loudness, etc. during the work. Music is known to heal the stress of a person. This project is a promising approach to providing personalized music recommendations to users. Collaborative filtering allows the system to recommend music based on the listening preferences of similar users, while k-means clustering algorithm helps in grouping similar songs together. The system can also handle new users and items using a cold start strategy. However, the effectiveness of the system depends heavily on the quality and size of the dataset used for training. In addition, the system may struggle to recommend niche or lesser-known music genres that have few ratings and limited user data. Thus, the work is very useful and efficient approach to music recommendation that has the potential to improve user satisfaction and engagement with music platforms.

V. ACKNOWLEDGEMENT

The authors are extremely thankful to Institute of Engineering & Management along with University of Engineering and Management, Kolkata for supporting in each and every step to carry out the research activities.

REFERENCES

[1] Kim, Dongmoon, Kun-su Kim, Kyo-Hyun Park, Jee-Hyong Lee, and Keon Myung Lee. "A music recommendation system with a dynamic k-means clustering algorithm." In Sixth international conference on machine learning and applications (ICMLA 2007), pp. 399-403. IEEE, 2007.

[2] Cano, Pedro, Markus Koppenberger, and Nicolas Wack. "Content-based music audio recommendation." In Proceedings of the 13th annual ACM international conference on Multimedia, pp. 211-212. 2005.

[3] Srifi, Mehdi, Ahmed Oussous, Ayoub Ait Lahcen, and Salma Mouline. "Recommender systems based on collaborative filtering using review texts—a survey." Information 11, no. 6 (2020): 317.

[4] Logan, Beth. "Music Recommendation from Song Sets." In ISMIR, pp. 425-428. 2004.

[5] Paul, Dip, and Subhradeep Kundu. "A survey of music recommendation systems with a proposed music recommendation system." In Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018, pp. 279-285. Springer Singapore, 2020.

[6] Pichl, Martin, Eva Zangerle, and Günther Specht. "Improving context-aware music recommender systems: Beyond the pre-filtering approach." In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pp. 201-208. 2017.

[7] Afoudi, Yassine, Mohamed Lazaar, and Mohammed Al Achhab. "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network." Simulation Modelling Practice and Theory 113 (2021): 102375.

[8] Velankar, Makarand, and Parag Kulkarni. "Music recommendation systems: overview and challenges." Advances in Speech and Music Technology: Computational Aspects and Applications (2022): 51-69.

[9] Pichl, Martin, Eva Zangerle, and Günther Specht. "Towards a context-aware music recommendation approach: What is hidden in the playlist name?." In 2015 IEEE international conference on data mining workshop (ICDMW), pp. 1360-1365. IEEE, 2015.

[10] Naik, Deepali, Varunraj Tipugade, Pranav Kulkarni, Aditi Shinde, and Ashootosh Pawar. "A Scientometric Analysis to Study Research Trends in Music Recommendation System using Collaborative Filtering and Semantic Analysis." In 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), pp. 1-6. IEEE, 2023.